Design Alternatives for Ultraperformance Parallel Computers

J.T. Schwartz

Courant Institute, N.Y.U.
251 Mercer Street
New York, N.Y. 10012

Ultracomputer Note #76

May, 1984

During the past two years, supercomputer developments in the U.S. and abroad have formed the subject of an intensive round of discussions in which industry, government, and universities have all participated. The recent news that the Hitachi 810 and Fujitsu VP200 vector computers outperform the CRAY XMP by a speed factor of roughly 2 (see Raul Mendez, The Japanese Supercomputer Challenge, parts I and II, SIAM News, Jan. and March, 1984) underscores the significance of these discussions. It is plainly time for the U.S. to look to its competitive position in this strategic area, maintenance of which will require extensive activity by industry, universities, and laboratories as the computing art advances into a new era of parallel supercomputing. Several major Federal initiatives which aim to strengthen U.S. ability to develop, manufacture, and use superspeed computers have just been established, notably the DARPA strategic computing initiative and the new NSF supercomputer access program. Significant though smaller scale efforts by DOE and other Federal agencies also exist and may well be enlarged in the near future.

By now, researchers (mostly at universities) studying superspeed computation have proposed over seventy designs for parallel machines. Since full experimental development of any one of these designs is likely to consume 5 to 20 million dollars or more, it is clear that only a small subset of this very large space of alternatives can be realized as large-scale systems. The accuracy with which U.S. industry and key Federal sponsoring agencies are able to discern those architectures likely to be of major future significance and to avoid 'dry holes' will therefore be crucial for development over the next few years.

This article will attempt to clarify the technical options available to parallel supercomputer development by listing various principal technical alternatives which characterize the design of such machines. Most of the parallel architectures that have been proposed occupy easily definable positions in the 'design space' spanned by these 'dimensions.' Various machines typifying some of the many possibilities belonging to this space will be noted.

The 'dimensions' of parallel supercomputer design can be listed, roughly in order of diminishing importance, as follows:

(1) *SIMD vs. MIMD* - A SIMD (single-instruction-multiple-data) machine consists of multiple processors (or of a few very fast pipelined processing elements), all of which are fed synchronously by a single instruction stream, but which operate in parallel on independent streams of data. Vector machines like the CRAY I or Hitachi 810, and synchronous array machines like the Illiac IV and NASA/Goodyear MPP typify this class. An MIMD computer consists of multiple processors (or fast timesliced processing elements) driven by independent instruction streams, and capable of branching independently, but also able to pass data and synchronization signals between processors, perhaps via a shared memory. The Denelcor HEP, Maryland ZMOB, Cal Tech Homogeneous Hypercube, and numerous other university machines belong to this class. SIMD machines are generally more efficient for applications characterized by very regular patterns of processing, in part because their 'lockstep' mode of operation can eliminate much of the message-passing overhead needed when all the processors of an SIMD machine must proceed through a computation in close synchrony. On the other hand, it is hard to make SIMD computers deal effectively with highly 'branched' code, since branching can only be emulated by temporarily disabling some of the processors in the parallel array, so that several successive branches reduce effective computational power drastically. MIMD processing arrays are more robust in this regard, since they are capable of executing not only innermost loops, but even complex branch-filled loops, in parallel.

(2) *Coarse-grained versus fine-grained* - The architect designing a parallel computer can decide to compose it either of the fastest sequential processors available at a given moment (as in the Livermore S-1), or of the most cost-effective single-chip processors available, or of the largest possible number of single bit processors, as in the NASA/Goodyear MPP. The decision to use a relatively small number of very high speed processors is justifiable by the argument that even largely parallelizable code will have significant serial sections, so that high performance in serial code will be important for sustained high performance. Its polar opposite, use of many single bit processors, can be justified by a desire to push parallel execution to its limits, and by the surprising level of arithmetic power per square millimeter of silicon which one-bit processors attain. Of the many arguments which would support use of an array of powerful single-chip microprocessors, the most obvious is the cost effectiveness of using a mass-produced item whose substantial development costs will have already been amortized.

(3) *Choice of interprocessor connection scheme* - Since the processors used in a large parallel array are apt to be relatively standard, most of the architectural novelty of such a machine will relate to the communication network chosen to allow close and effective cooperation between its processors. The communication scheme that a machine architect choses is apt to correlate strongly with the size of processing elements on which he has

fixed. If his parallel machine is built from a relatively small number of very high speed processors, it will be feasible and appropriate to connect them by a full crossbar, to achieve maximally close coupling. For substantially larger numbers of processors, this becomes infeasible, making it necessary to use more parsimonious communications networks. At the extreme limit of very large numbers of single-bit processors, communications nets that lay out well on the two-dimensional surface of a silicon chip will be preferred. This suggests use of two dimensional nearest-neighbor connections, either in a rectangular array (as in the NASA/Goodyear MPP) or in some more flexible, perhaps hexagonal scheme, for example that proposed in L. Snyder's 'Blue Chip' design. Another often-considered possibility is to use some tree-like interconnection scheme. For use in machines based on intermediate numbers of processors there have been devised various networks which grow only logarithmically with increasing numbers of processing elements, which nevertheless permit full point-to-point communication in a logarithmic number of steps. These include the hypercube connection (used in the Cal. Tech Homogeneous hypercube machine), the 'cube-connected cycles' network, and the perfect shuffle, omega, or banyan network proposed for the University of Texas TRAC machine and the NYU 'Ultracomputer'. One interesting design based upon single-bit processors, the Thinking Machines Corp. 'Connection Machine,' also makes use of a hypercube connection network.

(4) *Visible vs.'under-the-covers'connection net* - To emulate the action of a full crossbar, networks of the hypercube or banyan type need to move data through logarithmically many stages. If the direct network connections that do this are exposed to the system users, they can in some cases be exploited very effectively to perform certain key operations at maximum possible speed. However, this complicates programming of a parallel assemblage, which can be greatly simplified by hiding the lowest level network details under some type of microcoded cover and presenting the programmer with an appropriate interface in which information passes homogeneously either from any processor to any other, or between processors and some form of shared memory.

(5) *Circuit versus packet switching in the communication net* - Whatever the physical scheme used to pass signals along the communication net may be, the term 'packet switching' should be taken to describe an assemblage of processors thought of as retaining some fixed configuration while sharing work between them is a pattern which varies, perhaps gradually, from moment to moment. The contrasting term 'circuit switching' applies to a processor assemblage which is thought of as passing through a more discrete sequence of major phases, during each of which the role played by individual processors remains relatively stable. In this second view, change of computational phase is marked by 'reconfiguration,' i.e. by major shifts in the roles that individual processors play, by changes in interprocessor

communication pattern, and by interprocessor transmission of major blocks of data rather than short messages. As its name suggests, the Texas Reconfigurable Array Computer emphasizes this latter notion. However, packet switching designs are more numerous than circuit switching designs.

(6) *Specialization of architecture to an anticipated programming style and/or application* - If a parallel assemblage is intended to support some particular programming style, or to be heavily used for some special application, it may be appropriate to specialize its architecture correspondingly. Today's image processing systems illustrate this point, as do serial computers, such as the Symbolics Corp. LISP machine, which are designed to support particular high level languages very efficiently. The NASA/Goodyear MPP parallel array processor, whose design emphasizes very high speed image processing applications, is a larger-scale example. Various designs for large-scale 'dataflow' computers furnish additional examples. 'Dataflow' programming is organized as a set of fully parallel Petri-net related concepts. A program is regarded as an assemblage of 'operators,' all poised to fire as soon as they receive all their logical inputs. As soon as it fires, each such operator transmits its output to further operators, for which it becomes one of several possible inputs.

The innermost interpretation cycle for such a language therefore consists of various operand-routing, instruction dispatching, and instruction execution activities, many of which can be overlapped. As compared to a general purpose parallel assemblage, a cleverly designed layout of dataflow operation nodes in memory and effective instruction dispatch hardware can probably double or triple the efficiency with which a dataflow language executes, while adding only minimally to the total hardware cost of a parallel system.

This typifies the speed gains achievable by architectural specialization of parallel machines. Other architectural opportunities of this same sort are as follows: if we know the logical 'topology' of an intended application, i.e. the pattern in which information must pass between the processing subactivities of which the application is composed, and if the structure of the parallel array communication net is exposed, then it may be possible to devise some particularly efficient layout of code and data in the processing units of the parallel array. In a parallel processor of message-passing type, specialized message transmission, receipt, and buffering hardware may improve efficiency at very modest cost.

(7) *Balance among basic computational resources* - An architect attempting to design a serial computer must bring the ratios of memory to processing power and processing power to I/O capability into effective balance. These same considerations apply in the parallel case, but are further complicated by the need to balance additional ratios, including processing power to data motion power, and short-message passing capability to large data-block-

moving capability. The effectiveness of an architecture can then be rated by assessing its flexibility and by forming the ratio of average performance attained, in its intended applications, over the raw computational and interprocessor communication power that it embodies.

Conclusions

If we allow for even three choices along each of the seven-odd design dimensions reviewed above, there result some 2,187 conceivable parallel processor designs. Of these, more than seventy have already been proposed by the many research groups that have been attracted to this very active area; more designs are doubtless germinating. If the U.S. position in this area is not to decay further, the government agencies and industrial groups which command the resources required to mount significant development efforts must now choose wisely among the many alternatives offered. Systematic exploration of the most robust, general purpose, and fully scalable designs is what is first wanted. A design can be considered *robust* if it achieves high hardware utilization over a wide range of potential applications and makes effective use of hundreds of highly efficient parallel algorithms which have already been found by the rapidly growing community of researchers interested in parallel computation. A general purpose design can be defined as one which allows fully dynamic distribution of tasks between processing elements in a manner that avoids all memory location and message passing bottlenecks; is multiprogrammable; and supports high performance implementation of parallel versions of all major languages, including not only the established scientific computation languages, but also symbolically oriented set-theoretic languages, parallel variants of LISP, etc. A design is *scalable* if it extends without major changes to very large assemblages which could reach computation rates several hundred times larger than today's CRAY, Fujitsu, or Hitachi supercomputers.

Finally, parallel computer designers must be challenged to minimize the number of separate components needed to construct the machines they propose, to find effective VLSI implementations of these components, to address the complex packaging, cooling, and reliability problems sure to arise in large machine construction, to design peripheral devices which can handle the enormous volume of data which these machines will generate, and to create the operating and compiling systems which will make it possible to use these revolutionary new machines effectively.